LAVA + OpenQA = Automated, Continuous System Testing

Laurence Urhegyi & James Thomas

Introductions

- Laurence Urhegyi
  - Project Manager
  - laurence.urhegyi@codethink.co.uk

- James Thomas
  - Senior Engineer
  - james.thomas@codethink.co.uk

Both have a wealth of experience in the Automotive Industry, and are passionate about FOSS

Contents

- Long Term Maintainability
  - What is this and why do we care?
- What is the problem?
  - Upstream / downstream gap for LTM
- How we can help towards solving it?
  - Start testing in upstream directly
  - Heavy automation of testing
- What have we used?
  - LAVA
  - OpenQA
- What we have done?
  - Case study 1: openQA in GNOME upstream
  - Case study 2: LAVA/openQA in kernel testing
- Future plans

The problem

Why Long Term Maintainability?

Many software systems / devices / products are intended to function over a long-term lifetime.

They rely on open-source components.

Software complexity is increasing.

More features = more loc = more bugs = more testing is needed.

The cost and complexity of maintaining such systems will continue to grow over time.

Organisations need to be mature enough to handle increasing complexity and volume, whilst remaining competitive.

Don’t you mean Long Term Maintenance?

- Well, yes and no…
- Typical scenario:
  - Organisations rely on components (such as the kernel) that are supported for less time than the lifecycle of their product
  - Backporting from upstream is required for updates
  - Full system upgrades are rare or avoided
- The status quo is to ‘keep systems on life support’

We can do better…

Long Term Maintainability (again)

- Systems designed from the outset to:
    - Be as close as upstream as possible
        - Integrate new upstream with minimal overhead, reduce local patch carrying
    - Have robust deployment mechanisms in place, to roll-back if something goes wrong (atomic upgrades)
    - Have processes that allow you to know how to repeat a build done in the past, and also keep track of every component version deployed
    - **Have robust testing pipelines that makes you comfortable to update**

Testing, testing, one two, one two

- Everyone wants the same thing: stable software, including the latest features, for the long term

- To achieve this you need to have confidence in your software and confidence in the software you depend on
- This has to come from testing

Credit: SoundCloud

What do upstream propose?

- Always use the latest!
- http://kroah.com/log/blog/2018/01/06/meltdown-status/
- Makes complete sense… BUT

What downstream says

- Kernel is support for only 2 / 6 years
- We need to support this product for 10 years
- We do not want to support own own kernel fork
- We are not confident enough upgrading to next kernel version is not going to cause regressions!

- Not every organisation is willing / able to commit resources for testing that allows for such long term maintenance
- https://lore.kernel.org/linux-arm-kernel/YCzknUTDytY8gRA8@kroah.com/

Mind the gap!

- Testing modern software platforms is extremely complex and time consuming
- Companies often miss out on the latest fixes and features in Linux, for fear of the time-consuming manual regression tests required

(CC BY-SA 3.0) Wikipedia image

What we can do to help?

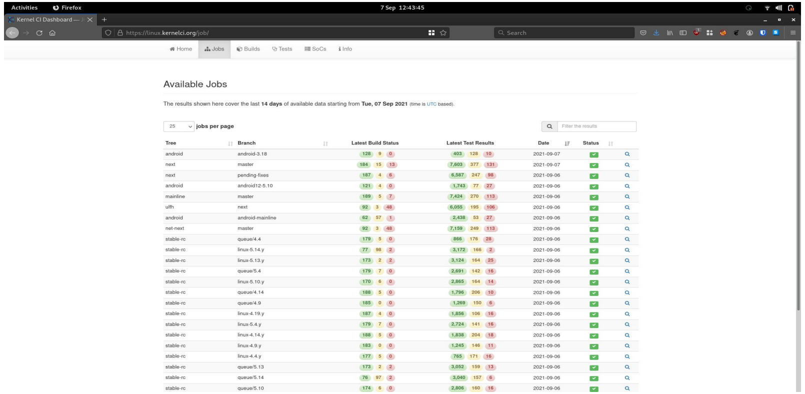
Automated, continuous embedded system testing FTW!

- Increased automation improves robustness and transparaceny of tests
- Reduce manual efforts = increased testing capacity
  - Machines can be utilised 24 hours per day
- Catch major errors more easily, and earlier
  - Eventually increase focus on more complex testing
- This is not new: https://security.googleblog.com/2021/08/linux-kernel-security-done-right.html

Intro to the tooling:
LAVA & openQA

LAVA

- LAVA “continuous integration system for deploying operating systems onto physical and virtual hardware for running tests”
- Hardware orchestration for physical devices
- Central server that dispatches jobs to clients
- Used by KernelCI (https://linux.kernelci.org/job/) to test new kernel versions



Available Jobs

The results shown here cover the last **14 days** of available data starting from **Tue, 07 Sep 2021** (time is UTC based).

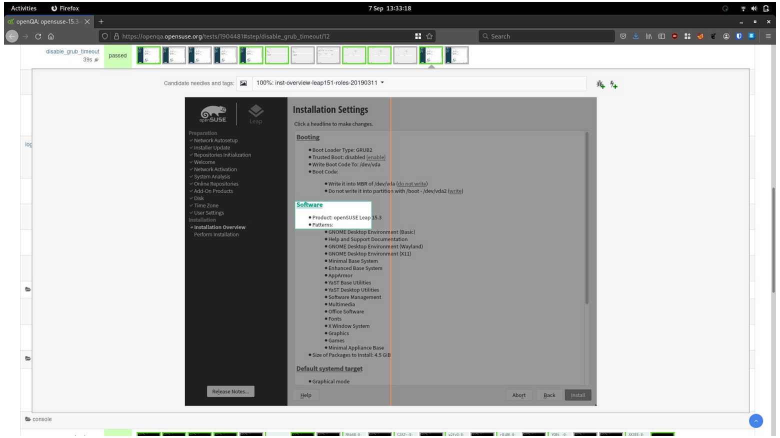
25 **jobs per page**

Filter the results

Tree	Branch	Latest Build Status	Latest Test Results	Date	Status	
android	android-3.18	128 9 0	403 128 10	2021-09-07	✓	🔍
next	master	184 15 13	7,603 377 131	2021-09-07	✓	🔍
next	pending-fixes	187 4 6	6,587 247 98	2021-09-06	✓	🔍
android	android12-5.10	121 4 0	1,743 77 27	2021-09-06	✓	🔍
mainline	master	189 5 7	7,424 270 113	2021-09-06	✓	🔍
ulfh	next	92 3 48	6,055 195 106	2021-09-06	✓	🔍
android	android-mainline	62 57 1	2,438 53 27	2021-09-06	✓	🔍
net-next	master	92 3 48	7,159 249 113	2021-09-06	✓	🔍
stable-rc	queue/4.4	179 5 0	866 176 28	2021-09-06	✓	🔍
stable-rc	linux-5.14.y	77 98 2	3,172 166 2	2021-09-06	✓	🔍
stable-rc	linux-5.13.y	173 2 2	3,124 164 25	2021-09-06	✓	🔍
stable-rc	queue/5.4	179 7 0	2,691 142 16	2021-09-06	✓	🔍
stable-rc	linux-5.10.y	170 6 0	2,865 164 14	2021-09-06	✓	🔍
stable-rc	queue/4.14	188 5 0	1,796 206 10	2021-09-06	✓	🔍
stable-rc	queue/4.9	185 0 0	1,269 150 6	2021-09-06	✓	🔍
stable-rc	linux-4.19.y	187 4 0	1,856 106 16	2021-09-06	✓	🔍
stable-rc	linux-5.4.y	179 7 0	2,724 141 16	2021-09-06	✓	🔍
stable-rc	linux-4.14.y	188 5 0	1,838 204 18	2021-09-06	✓	🔍
stable-rc	linux-4.9.y	183 0 0	1,245 146 11	2021-09-06	✓	🔍
stable-rc	linux-4.4.y	177 5 0	765 171 16	2021-09-06	✓	🔍
stable-rc	queue/5.13	173 2 2	3,052 159 13	2021-09-06	✓	🔍
stable-rc	queue/5.14	76 97 2	3,040 157 6	2021-09-06	✓	🔍
stable-rc	queue/5.10	174 6 0	2,806 160 16	2021-09-06	✓	🔍

OpenQA

- OpenQA for testing *software* (UI)
- Tests based on screenshot comparisons (Needles)
- Central server to view test results, and to dispatch test jobs to workers
- Workers provide a specific machine type (e.g qemu-x86_64)
- Tests performed how the majority of users actually use the software



https://openqa.opensuse.org/tests/1904481#step/disable_grub_timeout/12

Candidate needles and tags: 100%: inst-overview-leap151-roles-20190311

Preparation
✓ Network Autosetup
✓ Installer Update
✓ Repositories Initialization
✓ Welcome
✓ Network Activation
✓ System Analysis
✓ Online Repositories
✓ Add-On Products
✓ Disk
✓ Time Zone
✓ User Settings
Installation
→ Installation Overview
Perform Installation

Installation Settings

Click a headline to make changes.

Booting

- Boot Loader Type: GRUB2
- Trusted Boot: disabled (enable)
- Write Boot Code To: /dev/vda
- Boot Code:
  - Write it into MBR of /dev/vda (do not write)
  - Do not write it into partition with /boot - /dev/vda2 (write)

Software

- Product: openSUSE Leap 15.3
- Patterns:
  - GNOME Desktop Environment (Basic)
  - Help and Support Documentation
  - GNOME Desktop Environment (Wayland)
  - GNOME Desktop Environment (X11)
  - Minimal Base System
  - Enhanced Base System
  - AppArmor
  - YaST Base Utilities
  - YaST Desktop Utilities
  - Software Management
  - Multimedia
  - Office Software
  - Fonts
  - X Window System
  - Graphics
  - Games
  - Minimal Appliance Base
- Size of Packages to Install: 4.5 GiB

Default systemd target

- Graphical mode

Help upstream 1: openQA and GNOME

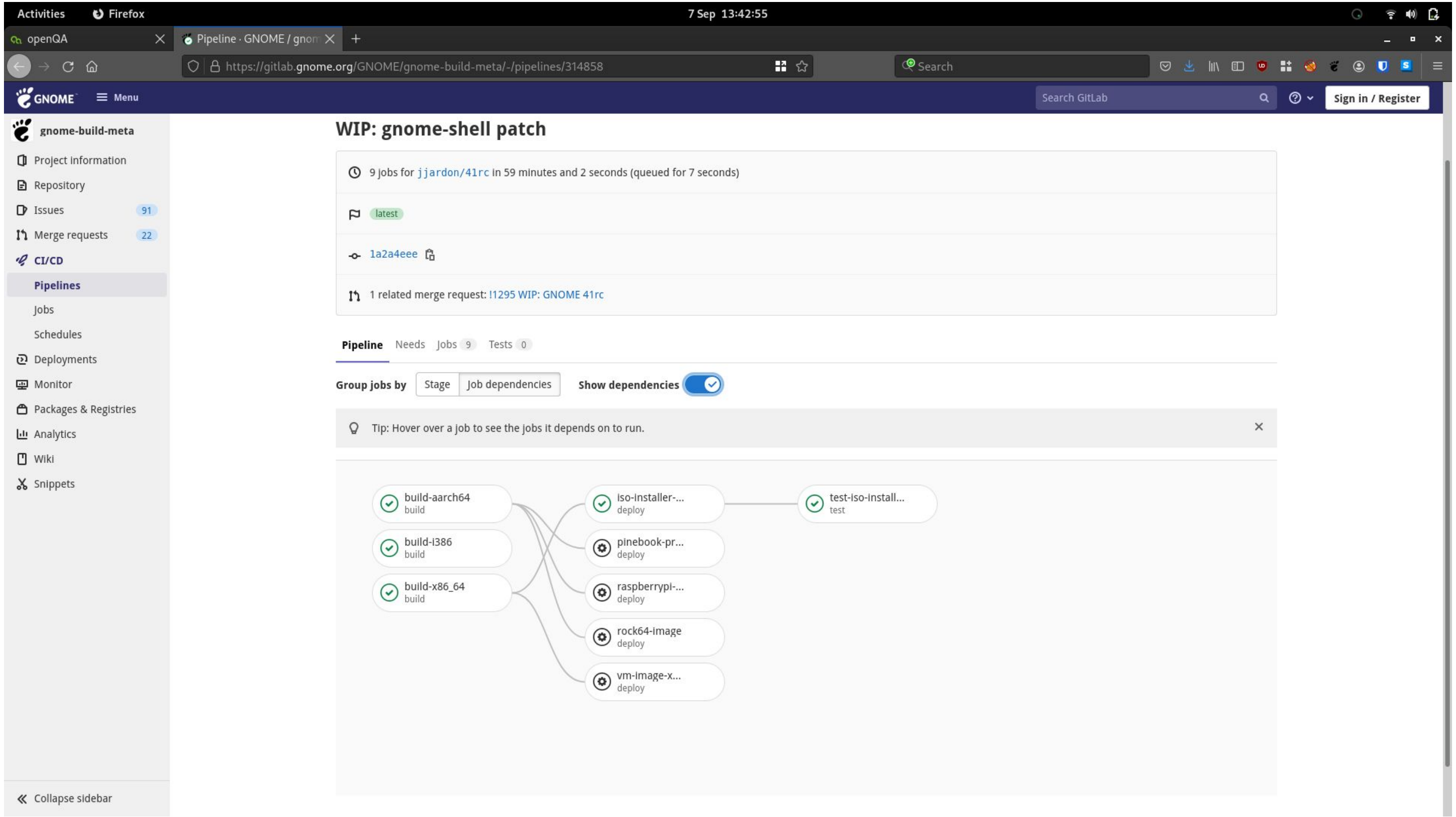
CI Integration

- Integrated into Gitlab CI pipeline
- Ad-hoc workers using the same build infrastructure, no “pets”, no dedicated test hardware needed
- Workers only need KVM

CI Integration

- Tests in the same git repo
- Different workers can run different tests if needed
- Needles in a separate repository
- Workers clone needles (OPENQA_NEEDLES_GIT, OPENQA_NEEDLES_SHA)
- Workers then register with OpenQA with a unique machine type (e.g qemu-x86_64-12345) and start the test run previously built GnomeOS image

CI Integration



WIP: gnome-shell patch

9 jobs for jjardon/41rc in 59 minutes and 2 seconds (queued for 7 seconds)

latest

1a2a4eee

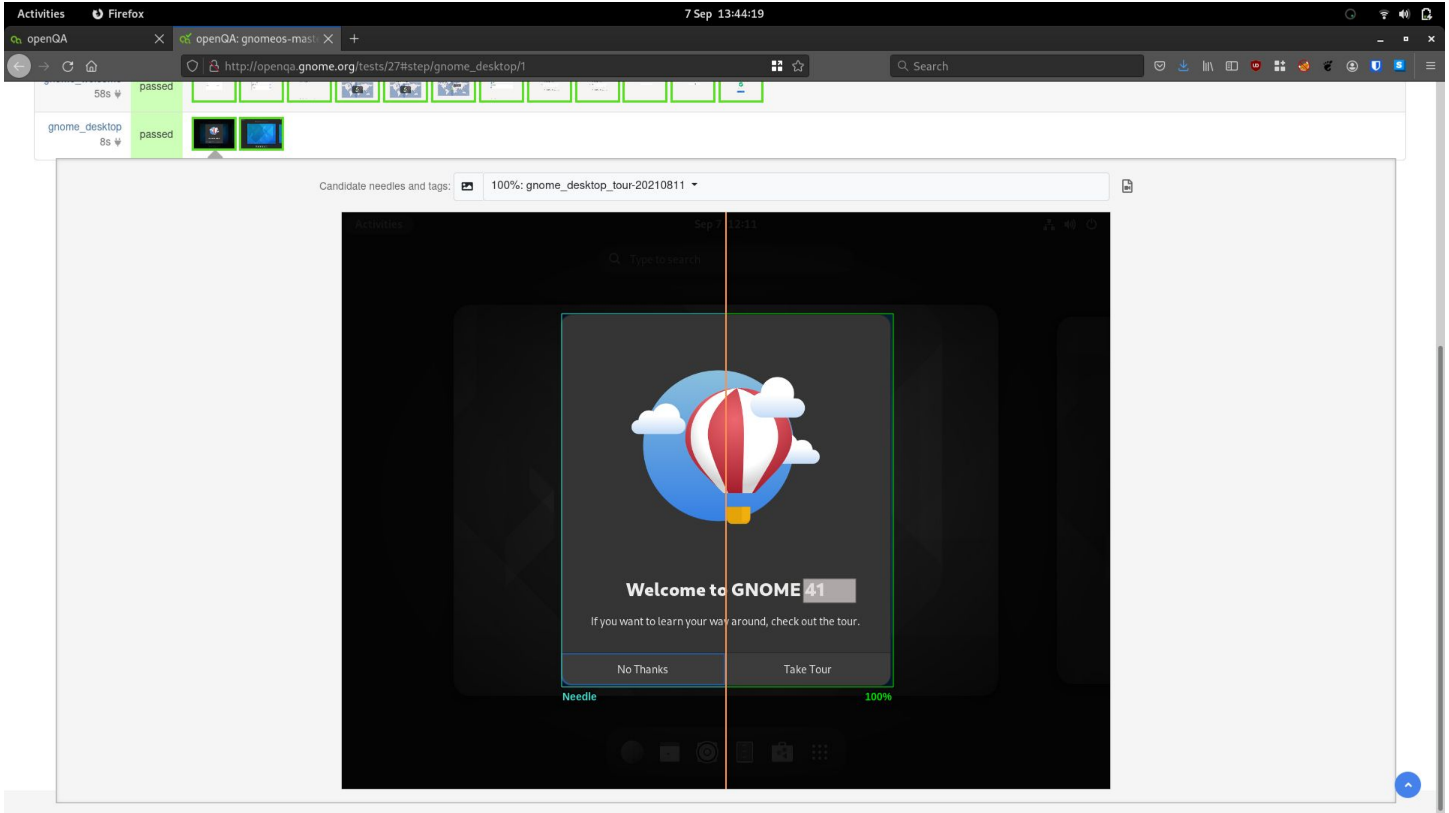
1 related merge request: !1295 WIP: GNOME 41rc

Group jobs by: Job dependencies

Show dependencies: on

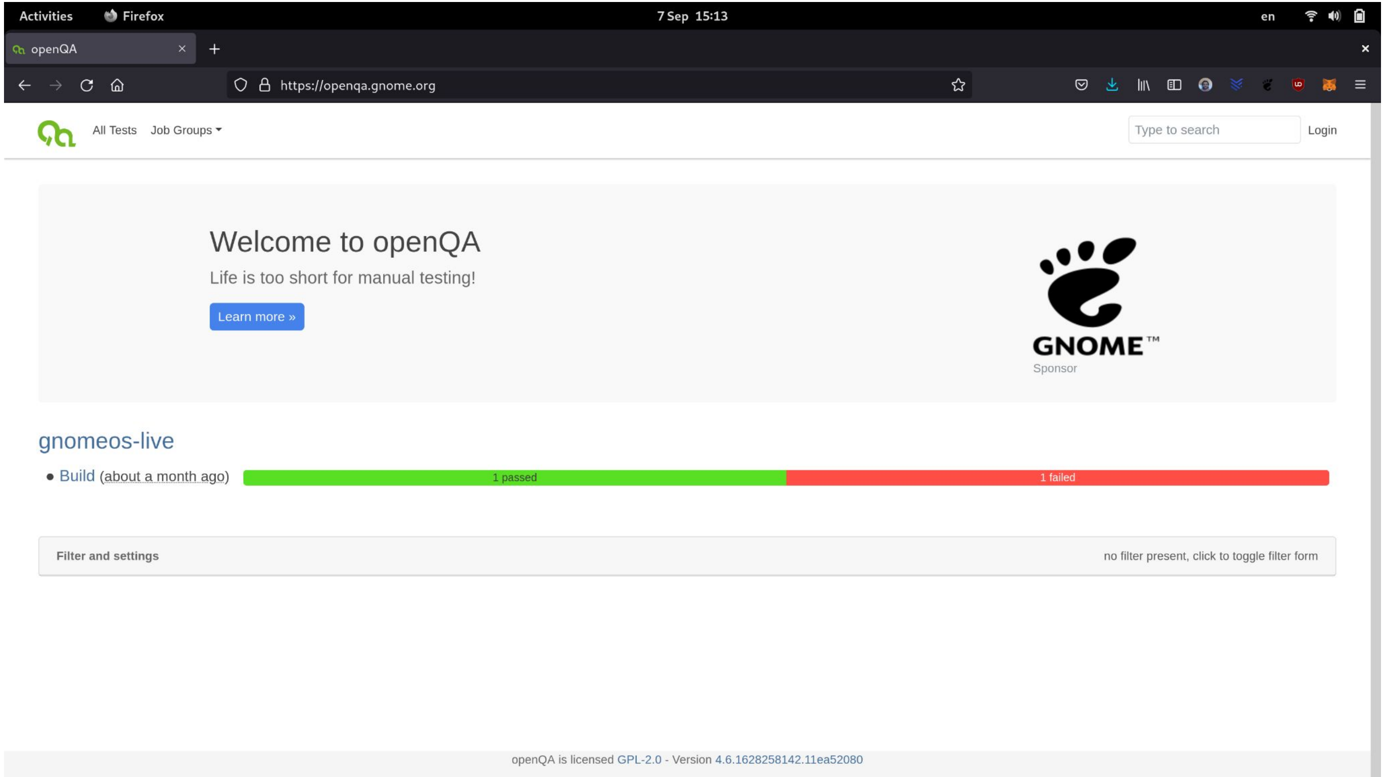
Tip: Hover over a job to see the jobs it depends on to run.

CI Integration



Activities
Firefox
7 Sep 13:44:19
openQA
openQA: gnomeos-mast
http://openqa.gnome.org/tests/27#step/gnome_desktop/1
Search
58s
passed
gnome_desktop
8s
passed
Candidate needles and tags:
100%: gnome_desktop_tour-20210811
Welcome to GNOME 41
If you want to learn your way around, check out the tour.
No Thanks
Take Tour
Needle
100%

Being used already: openqa.gnome.org



Welcome to openQA

Life is too short for manual testing!

gnomeos-live

- Build (about a month ago) 1 passed 1 failed

Filter and settings — no filter present, click to toggle filter form

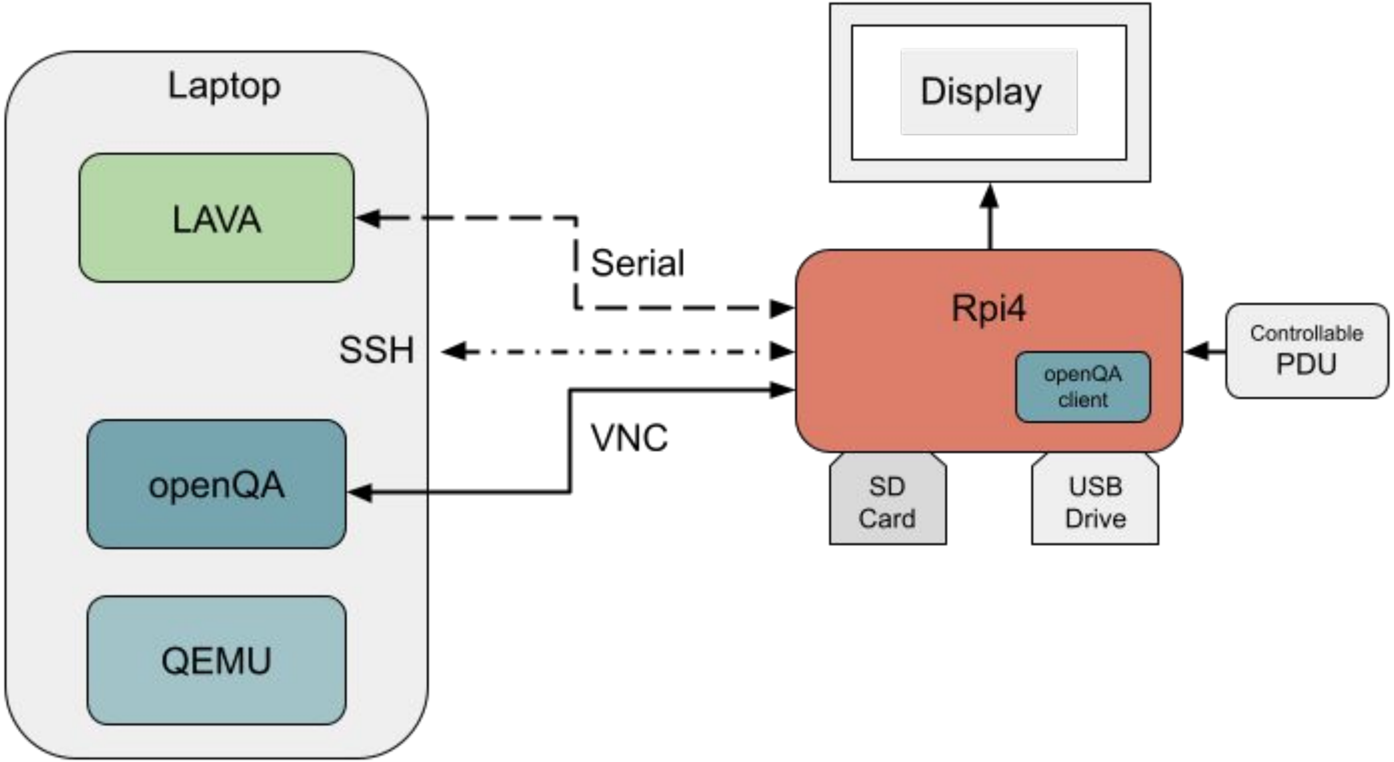
openQA is licensed GPL-2.0 - Version 4.6.1628258142.11ea52080

Help upstream 2:
LAVA & openQA & Linux kernel

LAVA and OpenQA

- Testing on hardware rather than emulators
- Utilise the existing KernelCI tests
- OpenQA testing to check for regressions (e.g in the graphics driver)
- Check out the blog post for details of the setup

Architecture



It’s being used!

- https://lava.qa.codethink.co.uk
- https://openqa.qa.codethink.co.uk/
- Positive feedback received https://lore.kernel.org/lkml/CAHk-=wjz+RhR8rr4rAZBPf-mxZXvn2RQe-XTQcL8X+HXBAFxBA@mail.gmail.com/

What next?

Codethink

Future Work

- More tests!
- Connect to KernelCI APIs
- Better Test/Needle synchronisation between workers and OpenQA
- Board deployment and testing added to Gitlab pipelines
- More backends! (e.g RDP)

Thanks for listening!

www.codethink.co.uk

Resources

- GNOME testing blog post
- https://openqa.gnome.org/
- https://gitlab.gnome.org/GNOME/gnome-build-meta/-/merge_requests/1251

- Kernel Testing blog post: LAVA + OpenQA
- https://lava.qa.codethink.co.uk
- https://openqa.qa.codethink.co.uk/